Unresolved Questions User Study		

1. How to use Xpath in Python? (8692)	XgbTree	Another option is py-dom-xpath, it works seamlessly with minidom and is pure Python so works on appengine.
		import xpath
		xpath.find('//item', doc)

	Answerbot	Sounds like an lxml advertisement in here. ;) ElementTree is included in the std library. Under 2.6 and below its xpath is pretty weak, but in 2.7+ much improved:
		import xml.etree.ElementTree as ET
		root = ET.parse(filename)
		result = ''
		for elem in root.findall('.//child/grandchild'):
		    # How to make decisions based on attributes even in 2.6:
		    if elem.attrib.get('name') == 'foo':
		        result = elem.text
		        break

	Ours	Sounds like an lxml advertisement in here. ;) ElementTree is included in the std library. Under 2.6 and below its xpath is pretty weak, but in 2.7+ much improved:
		import xml.etree.ElementTree as ET
		root = ET.parse(filename)
		result = ''
		for elem in root.findall('.//child/grandchild'):
		    # How to make decisions based on attributes even in 2.6:
		    if elem.attrib.get('name') == 'foo':
		        result = elem.text
		        break

2. What's the best way to duplicate fork() in windows? (23397)	XgbTree	You might also like using the processing module (http://pypi.python.org/pypi/processing). It has lot's of functionality for writing parallel systems with the same API as the threading module...
	Answerbot	Have a look at the process management functions in the os module. There are function for starting new processes in many different ways, both synchronously and asynchronously. I should note also that Windows doesn't provide functionality that is exactly like fork() on other systems. To do multiprocessing on Windows, you will need to use the threading module.
	Ours	fork() has in fact been duplicated in Windows, under Cygwin, but it's pretty hairy. The fork call in Cygwin is particularly interesting because it does not map well on top of the Win32 API. This makes it very difficult to implement correctly. See the The Cygwin User's Guide for a description of this hack.
3. Python version of PHP's stripslashes (13454)	XgbTree	You can obviously concatenate everything together: cleaned = stringwithslashes.replace("\\n","\n").replace("\\r","\n").replace("\\","") Is that what you were after? Or were you hoping for something more terse?
	Answerbot	Not totally sure this is what you want, but.. cleaned = stringwithslashes.decode('string_escape')

	Ours	It sounds like what you want could be reasonably efficiently handled through regular expressions:
		import re
		def stripslashes(s):
		    r = re.sub(r"\\(n|r)", "\n", s)
		    r = re.sub(r"\\", "", r)
		    return r
		cleaned = stripslashes(stringwithslashes)

4.Send file using POST from a Python script(150517)	XgbTree	Best thing I can think of is to encode it yourself. How about this subroutine?
		from urllib2 import Request, urlopen
		from binascii import b2a_base64
		
		def b64open(url, postdata):
		  req = Request(url, b2a_base64(postdata), headers={'Content-Transfer-Encoding': 'base64'})
		  return urlopen(req)
		
		conn = b64open("http://www.whatever.com/script.cgi", u"Liberté Égalité Fraternité")
		# returns a file-like object
		(Okay, so this code just sends POST-data. But you apparently want multipart-encoded data, as if you clicked an "Upload File" button, right? Well, it's a pretty straightforward combination of what I have here and the answers from the question you linked.)

	Answerbot	Best thing I can think of is to encode it yourself. How about this subroutine?
		from urllib2 import Request, urlopen
		from binascii import b2a_base64
		
		def b64open(url, postdata):
		  req = Request(url, b2a_base64(postdata), headers={'Content-Transfer-Encoding': 'base64'})
		  return urlopen(req)
		
		conn = b64open("http://www.whatever.com/script.cgi", u"Liberté Égalité Fraternité")
		# returns a file-like object
		(Okay, so this code just sends POST-data. But you apparently want multipart-encoded data, as if you clicked an "Upload File" button, right? Well, it's a pretty straightforward combination of what I have here and the answers from the question you linked.)

Unresolved Questions User Study		

	Ours	Best thing I can think of is to encode it yourself. How about this subroutine?
		from urllib2 import Request, urlopen
		from binascii import b2a_base64
		
		def b64open(url, postdata):
		  req = Request(url, b2a_base64(postdata), headers={'Content-Transfer-Encoding': 'base64'})
		  return urlopen(req)
		
		conn = b64open("http://www.whatever.com/script.cgi", u"Liberté Égalité Fraternité")
		# returns a file-like object
		(Okay, so this code just sends POST-data. But you apparently want multipart-encoded data, as if you clicked an "Upload File" button, right? Well, it's a pretty straightforward combination of what I have here and the answers from the question you linked.)

5. How do I edit and delete data in Django? (311188)	XgbTree & Answerbot	Say you have a model Employee. To edit an entry with primary key emp_id you do:
		emp = Employee.objects.get(pk = emp_id)
		emp.name = 'Somename'
		emp.save()
		to delete it just do: emp.delete()
		so a full view would be:
		def update(request, id):
		   emp = Employee.objects.get(pk = id)
		   #you can do this for as many fields as you like
		   #here I asume you had a form with input like <input type="text" name="name"/>
		   #so it's basically like that for all form fields
		   emp.name = request.POST.get('name')
		   emp.save()
		   return HttpResponse('updated')
		
		def delete(request, id):
		   emp = Employee.objects.get(pk = id)
		   emp.delete()
		   return HttpResponse('deleted')
		In urls.py you'd need two entries like this:
		(r'^delete/(\d+)/$','myproject.myapp.views.delete'),
		(r'^update/(\d+)/$','myproject.myapp.views.update'),

	Ours	To do either of these you need to use something called queries. check link below for really great documentation on that!, To Delete Data: b = ModelName.objects.get(id = 1) b.delete() This will delete the Object of the model w/ an ID of 1 To edit Data: b = ModelName.objects.get(id = 1) b.name = 'Henry' b.save() This will change the name of the Object of the model w/ an ID of 1 to be Henry

6. How can I normalize a URL in python (120951)	Xgbtree	Real fix in Python 2.7 for that problem
		Right solution was:
		# percent encode url, fixing lame server errors for e.g, like space
		 # within url paths.
		 fullurl = quote(fullurl, safe="%/:=&?~#+!$,;'@()*[]")
		For more information see Issue918368: "urllib doesn't correct server returned urls"

	Answerbot	Valid for Python 3.5:
		import urllib.parse
		urllib.parse.quote([your_url], "\./_-:")
		example:
		import urllib.parse
		print(urllib.parse.quote("http://www.example.com/foo goo/bar.html", "\./_-:"))

	Ours	Have a look at this module: werkzeug.utils. (now in werkzeug.urls)
		The function you are looking for is called "url_fix" and works like this:
		>>> from werkzeug.urls import url_fix
		>>> url_fix(u'http://de.wikipedia.org/wiki/Elf (Begriffsklärung)')
		'http://de.wikipedia.org/wiki/Elf%20%28Begriffskl%C3%A4rung%29'

7.How do I deploy a Python desktop application? (164137)	Xgbtree	Maybe IronPython can provide something for you? I bet those .exe/.dll-files can be pretty locked down. Not sure how such features work on mono, thus no idea how this works on Linux/OS X...
	Answerbot	You can distribute the compiled Python bytecode (.pyc files) instead of the source. You can't prevent decompilation in Python (or any other language, really). You could use an obfuscator like pyobfuscate to make it more annoying for competitors to decipher your decompiled source. As Alex Martelli says in this thread, if you want to keep your code a secret, you shouldn't run it on other people's machines. IIRC, the last time I used cx_Freeze it created a DLL for Windows that removed the necessity for a native Python installation. This is at least worth checking out.

Unresolved Questions User Study		
	Ours	You can distribute the compiled Python bytecode (.pyc files) instead of the source. You can't prevent decompilation in Python (or any other language, really). You could use an obfuscator like pyobfuscate to make it more annoying for competitors to decipher your decompiled source. As Alex Martelli says in this thread, if you want to keep your code a secret, you shouldn't run it on other people's machines. IIRC, the last time I used cx_Freeze it created a DLL for Windows that removed the necessity for a native Python installation. This is at least worth checking out.
8. Formatting a list of text into columns (171662)	Xgbtree	data = [("1","2"),("3","4")] print "\n".join(map("\t".join,data)) Not as flexible as the ActiveState solution, but shorter :-)
	Answerbot	This works it = iter(skills_defs) for i in it: print('{:<60}{}'.format(i, next(it, ""))) See: String format examples

	Ours	Two columns, separated by tabs, joined into lines. Look in itertools for iterator equivalents, to achieve a space-efficient solution.
		import string
		def fmtpairs(mylist):
		    pairs = zip(mylist[::2],mylist[1::2])
		    return '\n'.join('\t'.join(i) for i in pairs)
		print fmtpairs(list(string.ascii_uppercase))
		
		A more general solution, handles any number of columns and odd lists. Slightly modified from S.lott, using generators to save space.
		def fmtcols(mylist, cols):
		    lines = ("\t".join(mylist[i:i+cols]) for i in xrange(0,len(mylist),cols))
		    return '\n'.join(lines)

9. Setup Python environment on Windows (182053)	Xgbtree	Download the Python 2.6 Windows installer from python.org (direct link). If you're just learning, use the included SQLite library so you don't have to fiddle with database servers. Most web development frameworks (Django, Turbogears, etc) come with a built-in webserver command that runs on the local computer without Apache.
	Answerbot	Bundle: go with Activestate's Python, which bundles many useful win32-related libraries. It has no version 2.6 yet, but most code you'll find online refers to 2.5 and lower anyway. Database: any of the popular open-source DBs are simple to configure. But as John already suggested, for simple beginning stuff just use SQLite which already comes bundled with Python. Web server: depends on the scale. You can configure Apache, yes, but for trying simple things the following is a quite complete web server in Python that will also serve CGI scripts writte in Python:
	Ours	Download the Python 2.6 Windows installer from python.org (direct link). If you're just learning, use the included SQLite library so you don't have to fiddle with database servers. Most web development frameworks (Django, Turbogears, etc) come with a built-in webserver command that runs on the local computer without Apache.
10. How to read and write multiple files?(208120)	Xgbtree	You may find the fileinput module useful. It is designed for exactly this problem.

	Answerbot	I think what you miss is how to retrieve all the files in that directory. To do so, use the glob module.
		Here is an example which will duplicate all the files with extension *.txt to files with extension *.out
		
		import glob
		list_of_files = glob.glob('./*.txt') # create the list of file
		for file_name in list_of_files:
		  FI = open(file_name, 'r')
		  FO = open(file_name.replace('txt', 'out'), 'w')
		  for line in FI:
		    FO.write(line)
		  FI.close()
		  FO.close()

	Ours	import sys
		# argv is your commandline arguments, argv[0] is your program name, so skip it
		for n in sys.argv[1:]:
		    print(n) #print out the filename we are currently processing
		    input = open(n, "r")
		    output = open(n + ".out", "w")
		    # do some processing
		    input.close()
		    output.close()
		
		Then call it like:
		./foo.py bar.txt baz.txt

11. How can I remove a trailing	Xgbtree	I might use something like this:
		import os
		s = s.rstrip(os.linesep)
		I think the problem with rstrip("\n") is that you'll probably want to make sure the line separator is portable. (some antiquated systems are rumored to use "\r\n"). The other gotcha is that rstrip will strip out repeated whitespace. Hopefully os.linesep will contain the right characters. the above works for me.

Unresolved Questions User Study		
newline? (275018)	Answerbot	You may use line = line.rstrip('\n'). This will strip all newlines from the end of the string, not just one.

	Ours	And I would say the "pythonic" way to get lines without trailing newline characters is splitlines().
		>>> text = "line 1\nline 2\r\nline 3\nline 4"
		>>> text.splitlines()
		['line 1', 'line 2', 'line 3', 'line 4']

12 How to import a python file in python script more than once (276679)	Xgbtree	While Tom Ley's answer is the correct approach, it is possible to import a module more than once, using the reload built-in.
		module.py:
		print "imported!"
		
		>>> import module
		imported!
		>>> reload(module)
		imported!
		<module 'module' from 'module.pyc'>

	Answerbot	You most probably should not use import for what you are trying to do. Without further information I can only guess, but you should move the code in the module you import from the top level into a function, do the import once and than simply call the function from you loop.
	Ours	The import statement -- by definition -- only imports once. You can, if you want, try to use execfile() (or eval()) to execute a separate file more than once.

13. How to split a web address (286150)	Xgbtree	Dan is right: urlparse is your friend:
		>>> from urlparse import urlparse
		>>>
		>>> parts = urlparse("http://www.stackoverflow.com/questions/ask")
		>>> parts.scheme + "://" + parts.netloc
		'http://www.stackoverflow.com'
		>>> parts.path
		'/questions/ask'
		Note: In Python 3 it's from urllib.parse import urlparse

	Answerbot	Dan is right: urlparse is your friend:
		>>> from urlparse import urlparse
		>>>
		>>> parts = urlparse("http://www.stackoverflow.com/questions/ask")
		>>> parts.scheme + "://" + parts.netloc
		'http://www.stackoverflow.com'
		>>> parts.path
		'/questions/ask'
		Note: In Python 3 it's from urllib.parse import urlparse

	Ours	Use the Python urlparse module: https://docs.python.org/library/urlparse.html For a well-defined and well-traveled problem like this, don't bother with writing your own code, let alone your own regular expressions. They cause too much trouble ;-).
14. Python object deleting itself (293431)	Xgbtree	You don't need to use del to delete instances in the first place. Once the last reference to an object is gone, the object will be garbage collected. Maybe you should tell us more about the full problem.
	Answerbot	You don't need to use del to delete instances in the first place. Once the last reference to an object is gone, the object will be garbage collected. Maybe you should tell us more about the full problem.
	Ours	You don't need to use del to delete instances in the first place. Once the last reference to an object is gone, the object will be garbage collected. Maybe you should tell us more about the full problem.

15. How to validate IP address in Python? (319279)	Xgbtree	Don't parse it. Just ask.
		import socket
		
		try:
		    socket.inet_aton(addr)
		    # legal
		except socket.error:
		    # Not legal

	Answerbot	Don't parse it. Just ask.
		import socket
		
		try:
		    socket.inet_aton(addr)
		    # legal
		except socket.error:
		    # Not legal

Unresolved Questions User Study		

	Ours	I think this would do it...
		def validIP(address):
		    parts = address.split(".")
		    if len(parts) != 4:
		        return False
		    for item in parts:
		        if not 0 <= int(item) <= 255:
		            return False
		    return True

16. Read the last lineof the file (327776)	Xgbtree	f.seek(pos ,2) seeks to 'pos' relative to the end of the file. try a reasonable value for pos then readlines() and get the last line. You have to account for when 'pos' is not a good guess, i.e. suppose you choose 300, but the last line is 600 chars long! in that case, just try again with a reasonable guess, until you capture the entire line. (this worst case should be very rare)

	Answerbot	Um why not just seek to the end of the file and read til you hit a newline?.
		i=0
		while(1):
		    f.seek(i, 2)
		    c = f.read(1)
		    if(c=='\n'):
		        break

	Ours	f.seek(pos ,2) seeks to 'pos' relative to the end of the file. try a reasonable value for pos then readlines() and get the last line. You have to account for when 'pos' is not a good guess, i.e. suppose you choose 300, but the last line is 600 chars long! in that case, just try again with a reasonable guess, until you capture the entire line. (this worst case should be very rare)
17. Comparing dictionaries in Python (434353)	Xgbtree	You can simplify this by using a defaultdict. Calling __getitem__ on a defaultdict will return the "default" value. from collections import defaultdict d = defaultdict(int) print d['this key does not exist'] # will print 0 Another bit that you could change is not to call keys. The dictionary implements iter. It would be preferable to simply write: for k in d1:

	Answerbot	Yours is actually fine -- you could simplify it to
		for k in d1:
		    if d2.get(k, 0) < l:
		       print k, d2.get(k, 0)
		which is (to me) pythonic, and is pretty much a direct "translation" into code of your description.
		If you want to avoid the double lookup, you could do
		for k in d1:
		    val = d2.get(k, 0)
		    if val < l:
		        print k, val

	Ours	Yours is actually fine -- you could simplify it to
		for k in d1:
		    if d2.get(k, 0) < l:
		       print k, d2.get(k, 0)
		which is (to me) pythonic, and is pretty much a direct "translation" into code of your description.
		If you want to avoid the double lookup, you could do
		for k in d1:
		    val = d2.get(k, 0)
		    if val < l:
		        print k, val

18 Is there a label/goto in Python? (438844)	Xgbtree	I recently wrote a function decorator that enables goto in Python, just like that:
		from goto import with_goto
		
		@with_goto
		def range(start, stop):
		    i = start
		    result = []
		
		    label .begin
		    if i == stop:
		        goto .end
		
		    result.append(i)
		    i += 1
		    goto .begin
		
		    label .end
		    return result
		I'm not sure why one would like to do something like that though. That said, I'm not too serious about it. But I'd like to point out that this kind of meta programming is actual possible in Python, at least in CPython and PyPy, and not only by misusing the debugger API as that other guy did. You have to mess with the bytecode though.

Unresolved Questions User Study		
	Answerbot	Python offers you the ability to do some of the things you could do with a goto using first class functions. For example: … Granted, that isn't the best way to substitute for goto. But without knowing exactly what you're trying to do with the goto, it's hard to give specific advice.
	Ours	No, Python does not support labels and goto, if that is what you're after. It's a (highly) structured programming language.
19. No module named MySQLdb(454854)	Xgbtree	You need to use one of the following commands. Which one depends on what OS and software you have and use 1. easy_install mysql-python (mix os) 2.pip install mysql-python (mix os/ python 2) 3.pip install mysqlclient (mix os/ python 3) 4.apt-get install python-mysqldb (Linux Ubuntu, ...) 5.cd /usr/ports/databases/py-MySQLdb && make install clean (FreeBSD) 6.yum install MySQL-python (Linux Fedora, CentOS ...) For Windows, see this answer: Install mysql-python (Windows)
	Answerbot	if your python version is 3.5, do a pip install mysqlclient, other things didn't work for me
	Ours	You need to use one of the following commands. Which one depends on what OS and software you have and use 1. easy_install mysql-python (mix os) 2.pip install mysql-python (mix os/ python 2) 3.pip install mysqlclient (mix os/ python 3) 4.apt-get install python-mysqldb (Linux Ubuntu, ...) 5.cd /usr/ports/databases/py-MySQLdb && make install clean (FreeBSD) 6.yum install MySQL-python (Linux Fedora, CentOS ...) For Windows, see this answer: Install mysql-python (Windows)
20. How can I read system information in Python on OS X?(467600)	Xgbtree	I did some more googling (looking for "OS X /proc") -- it looks like the sysctl command might be what you want, although I'm not sure if it will give you all the information you need. Here's the manpage: … Also, wikipedia.
	Answerbot	You can get a large amount of system information from the command line utilities sysctl and vm_stat (as well as ps, as in this question.) If you don't find a better way, you could always call these using subprocess.

	Ours	The only stuff that's really nicely accesible is available from the platform module, but it's extremely limited (cpu, os version, architecture, etc). For cpu usage and uptime I think you will have to wrap the command line utilities 'uptime' and 'vm_stat'.
		I built you one for vm_stat, the other one is up to you ;-)import os, sys
		
		def memoryUsage():
		    result = dict()
		    for l in [l.split(':') for l in os.popen('vm_stat').readlines()[1:8]]:
		        result[l[0].strip(' "').replace(' ', '_').lower()] = int(l[1].strip('.\n '))
		    return result
		print memoryUsage()

21. Get rid of '\n' in Python(495424)	Xgbtree	"string \n".strip();
		or
		"string \n".rstrip();

	Answerbot	In Python 3, to print a string without a newline, set the end to an empty string: print("some string", end="")

	Ours	Get rid of just the "\n" at the end of the line:
		>>> "string \n".rstrip("\n")
		'string '
		Get rid of all whitespace at the end of the line:
		>>> "string \n".rstrip()
		'string'
		Split text by lines, stripping trailing newlines:
		>>> "line 1\nline 2 \nline 3\n".splitlines()
		['line 1', 'line 2 ', 'line 3']

22. How to convert a list of number to html in Python? (520881)	Xgbtree	I think you either want this:
		numbers = [100, 101, 102, 103]
		output = "<td>" + ", ".join(map(str, numbers)) + "</td>"
		or
		output = ""
		for number in numbers:
		    output += "<td>" + str(number) + "</td>"

	Answerbot	As pointed out in the comments, it's really hard to understand your question... but if you want to generate a bunch of table cells, each containing one of the numbers, use something like this: html = ''.join('<td>%d</td>' % n for n in numbers) Of course you can add in a class or other attribute to be applied to the table cells if you want.
	Ours	As pointed out in the comments, it's really hard to understand your question... but if you want to generate a bunch of table cells, each containing one of the numbers, use something like this: html = ''.join('<td>%d</td>' % n for n in numbers) Of course you can add in a class or other attribute to be applied to the table cells if you want.

Unresolved Questions User Study		
23. How can I find memory leaks in my Python program?(532346)	Xgbtree	Python's memory is managed by a garbage collector. In general, there shouldn't be a problem with memory leaking (definitely not for Python2.5 and above), unless you happen to be writing extension modules in C/C++. In that case, Valgrind (Blog post -http://bruynooghe.blogspot.com/2008/12/finding-memory-leaks-in-python.html) might be helpful. I found that this person - http://mg.pov.lt/blog/hunting-python-memleaks has used PDB and matplotlib to trace a memory leak. I hope this helps, I have no experience fixing Python memory leaks.
	Answerbot	Python's memory is managed by a garbage collector. In general, there shouldn't be a problem with memory leaking (definitely not for Python2.5 and above), unless you happen to be writing extension modules in C/C++. In that case, Valgrind (Blog post -http://bruynooghe.blogspot.com/2008/12/finding-memory-leaks-in-python.html) might be helpful. I found that this person - http://mg.pov.lt/blog/hunting-python-memleaks has used PDB and matplotlib to trace a memory leak. I hope this helps, I have no experience fixing Python memory leaks.
	Ours	Generally, failing to close cursors is one of the most common kinds of memory leaks. The garbage collector can't see the MySQL resources involved in the cursor. MySQL doesn't know that the Python side was released unless the close() method is called explicitly. Rule of thumb. Open, use and close cursors in as short a span of code as you can manage.
24 Python - replace random items in column(12035394)	Xgbtree	I would strongly suggest reading a python primer, the way your problem solved can be done in two steps 1. read items from file - reference 2. use math.random() to change random string- reference by know how to do these points you can easily achieve what you intend to do.
	Answerbot	Use this to generate a random string import os random_string = os.urandom(string_length) To loop over a file line by line, do with open('file') as fd: for line in fd: # do stuff No need to close the file handle. use split to well, split on whitespace and place the result in an array (indexing starts at 0) Read more at docs.python.org. Please update your question with some code when you have gotten that far... Good luck
	Ours	I would strongly suggest reading a python primer, the way your problem solved can be done in two steps 1. read items from file - reference 2. use math.random() to change random string- reference by know how to do these points you can easily achieve what you intend to do.

25. Python: how to adjust x axis in matplotlib ? (12109648)	XgbTree	Look at the docs :
		    xlocs, xlabs = plt.xticks()
		put in xlocs your range, and in xlabs what you want to display. then:
		    plt.xticks(xlocs, xlabs)

	Answerbot	The size of the plot can be changed by setting the dynamic rc settings of Matplotlib. These are stored in a dictionary named rcParams. The size of the plot figure is stored with the key figure.figsize.

	Ours	It sounds like you want to changes the limits of the plotting display - for that use xlim (and ylim for the other axis). To change the xticks themselves is provided in the answer by @fp. Show below is an example using without/with xlim:
		    import pylab as plt
		    plt.subplot(2,1,1)
		    plt.hist(X,bins=300)
		    plt.subplot(2,1,2)
		    plt.hist(X,bins=300)
		    plt.xlim(0,100)
		    plt.show()